

Chapter 7

Creating and Managing Azure Resources

**MICROSOFT EXAM OBJECTIVES COVERED
IN THIS CHAPTER:**

DESCRIBE CORE SOLUTIONS AND MANAGEMENT TOOLS ON AZURE

✓ Describe Azure management tools

- Describe the functionality and usage of the Azure Portal, Azure PowerShell, Azure CLI, Cloud Shell, and Azure Mobile App





Azure is a very rich, complex offering. As the number of services and resources rises, so does the complexity of deploying and managing resources. Fortunately, Azure offers multiple methods for managing the resources in your Azure environment. In most cases, you can use one of several management tools for the same task, enabling you to choose a tool that offers the appropriate balance between ease and functionality.

This chapter explores the management tools available in Azure and helps you put them to use to gain some experience in creating and managing your own Azure resources.

Azure Management Tools

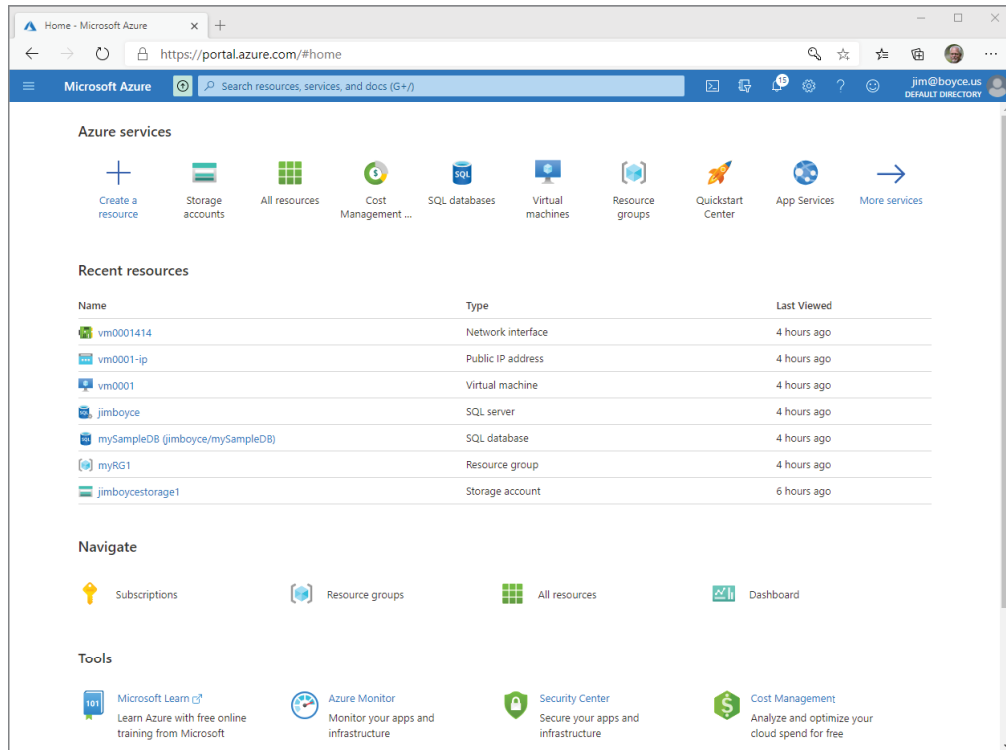
It is easy to deploy a single VM using the Azure portal. You choose some options from a few controls on a web page and soon your VM is up and running. But what if you need to deploy 50 VMs? What if you have internal line-of-business apps that you have to integrate into the deployment process? In those scenarios, a web portal is far from optimal. That's why it is important to choose the **right management option** for each scenario. Let's examine these options, starting with the Azure portal.

Azure Portal

The Azure portal is a **web interface** that enables you to **view, create, and manage** Azure resources and services. The Azure portal provides access to almost every aspect of Azure, but as hinted at earlier, it is not always the best management solution for a given scenario. Nevertheless, the Azure portal offers the **simplest solution** for managing individual resources and services. Figure 7.1 shows the Azure portal.

When choosing a management tool for Azure, the most important considerations for the Azure portal are **ease of use and visualization**. The Azure portal is easy to use because it offers a familiar web-based user experience. It also provides a wealth of **visualization tools and reports** for understanding your Azure environment and managing it. One benefit of the Azure portal is that, as a web-based tool, it is available on most devices with a browser. So, if you need to create a VM from an Android laptop or tablet, doing so is as easy as on a Windows device.

FIGURE 7.1 You can use the Azure portal to manage most Azure resources.

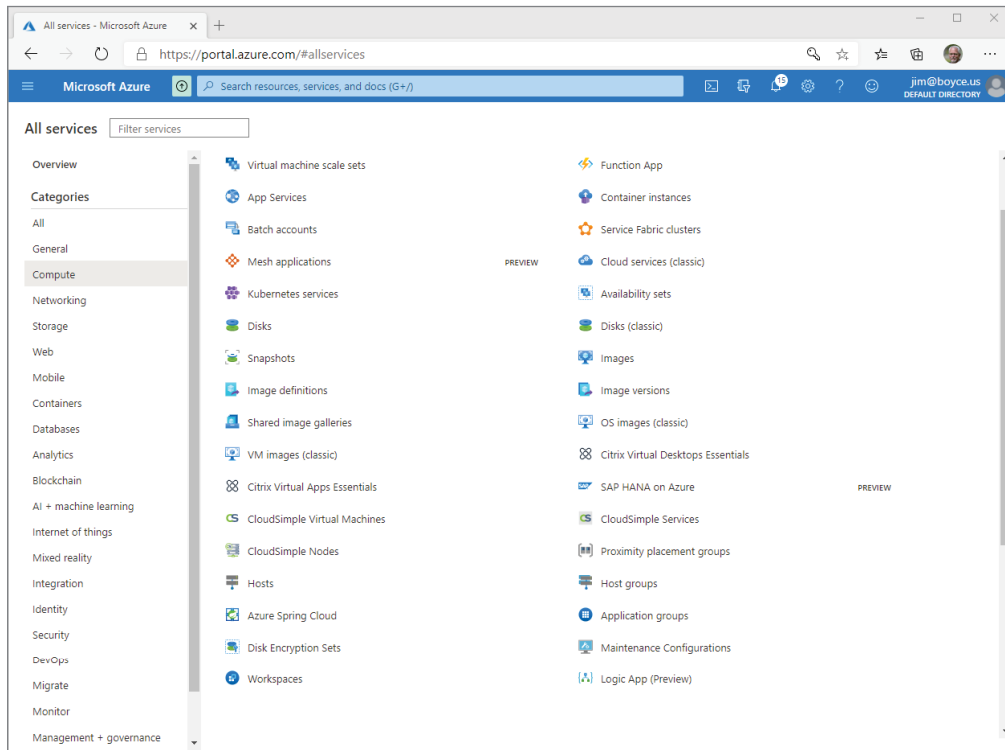


With changes happening in Azure all the time, the Azure portal might look somewhat different for you compared to the images in this book. The overall functionality is the same.

The home page of the Azure portal is simple, with common services shown at the top and recent resources listed below, along with navigation options and tools. Clicking a service in the Azure Services list at the top opens a page for that service. Clicking More Services opens a navigation pane at the left that lists service categories. If you want to work with VMs and related resources, for example, click Compute. The Azure portal then shows a list of all the Compute resources and services, as shown in Figure 7.2.

You might encounter several questions on the AZ-900 exam that ask what Azure portal *blade* you should use to manage specific Azure resources or access specific services, such as the Security Center. The Azure portal design has changed over time, but the items in the Azure portal menu have long been referred to as blades. The best way to prepare for these questions is to spend some time in the Azure portal, getting familiar with it and understanding which items in the Azure portal menu (navigation pane) give you access to which services.

FIGURE 7.2 The Compute category is selected in the Azure portal with all Compute resources and services listed.



You can configure the Azure portal to either dock the Azure portal menu or use it as a flyout menu. Click the Settings button in the portal to open the Portal Settings menu, where you can choose between Flyout and Docked.

Azure PowerShell

Azure PowerShell is a **scripting environment** that you can use to **execute commands** (called cmdlets, pronounced *command-lets*) that **perform management tasks** in Azure through the Azure REST API. You can perform essentially any task in Azure using Azure PowerShell. You can also combine cmdlets into scripts to execute multiple commands to orchestrate more complex tasks. These scripts can be quite simple or very complex, potentially deploying hundreds of resources and supporting services in a very short period.



Using Azure PowerShell scripts provides a repeatable method for automating specific tasks in Azure.

Azure PowerShell is available on Windows, Linux, Mac, and ARM. You can also access Azure PowerShell through the Azure Cloud Shell, discussed later in this chapter in the section “Azure Cloud Shell.” This is a key point to remember for the AZ-900 exam—in scenarios where it is not possible or feasible to run Azure PowerShell natively, you can use the Azure Cloud Shell to run Azure PowerShell.



You must install Azure PowerShell to use the tool natively on a device. Azure PowerShell Core is available as an open source project on GitHub. For more information, see <https://docs.microsoft.com/en-us/powershell/scripting/install/installing-powershell>.

Azure CLI

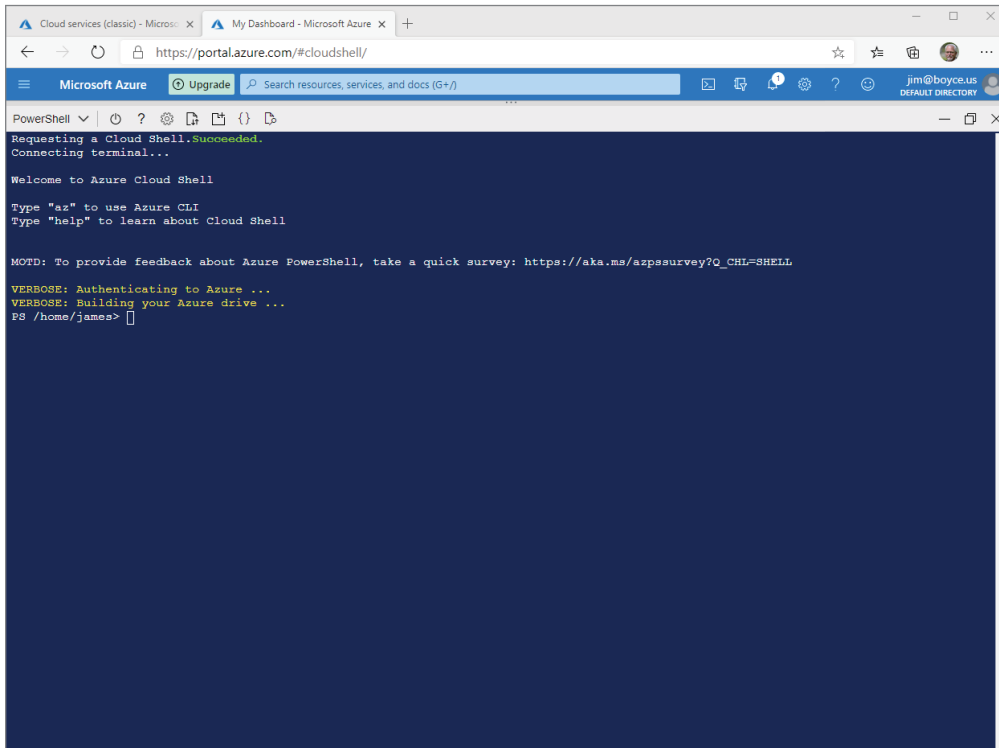
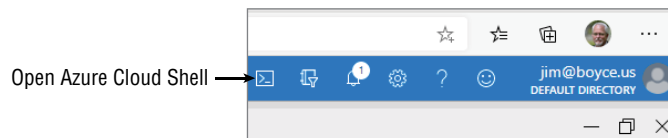
If you are experienced with Linux and proficient with the Bash shell and command language, you might prefer to use the Azure command-line interface (Azure CLI) instead of Azure PowerShell. Like Azure PowerShell, the Azure CLI is a command-driven scripting environment that also uses the Azure REST API to execute management tasks in Azure. It is nearly identical to Azure PowerShell in capability and function, running on Windows, Linux, and Mac. It simply uses a different command syntax with which you might be more familiar as a Bash shell user. Also, as with Azure PowerShell, you can run the Azure CLI in a web browser through the Azure Cloud Shell, giving you the capability to run the Azure CLI in scenarios where running it natively might not be possible or feasible.

Azure Cloud Shell

As described briefly earlier, Azure Cloud Shell is a web-based interface that enables you to run Azure PowerShell and Azure CLI commands and scripts. You can access Azure Cloud Shell at <https://shell.azure.com> (which redirects to <https://portal.azure.com/#cloudshell>, but shell.azure.com is easier to remember and type). On the resulting page you click either Bash or PowerShell depending on which environment you want to use. Figure 7.3 shows the Cloud Shell running PowerShell.



You can also access the Azure Cloud Shell through the Azure portal by clicking the Cloud Shell icon in the group of icons to the left of your username in the upper right corner of the browser. Figure 7.4 shows an example.

FIGURE 7.3 The Azure Cloud Shell running Azure PowerShell**FIGURE 7.4** You can quickly access the Cloud Shell from the Azure portal.

The key point to understand about the Azure Cloud Shell is that it enables you to run either Azure PowerShell or the Azure CLI in a web browser.

Azure Mobile App

The Azure Mobile App is available for Android and iOS devices and enables you to **manage Azure resources** from your **mobile device**. You can use the Azure Mobile App for one-off tasks such as monitoring health and status of resources, checking alerts, restarting web apps or VMs, and running Azure CLI or Azure PowerShell commands. Naturally, the Azure Mobile App isn't a great management solution when you need to perform more complex

tasks, but it nevertheless enables you to perform many Azure management functions from your mobile device. For example, if you need to reset a web app, you can open the Azure Mobile App and run the appropriate Azure CLI command to perform the reset.

Using ARM Templates

Azure Resource Manager (ARM) templates are discussed in earlier chapters, so you should have some understanding by now of their function. To summarize, ARM templates are **JSON files** that declare **infrastructure and configuration** for Azure resources, enabling you to deploy resources in a well-defined, repeatable way. Rather than use ARM templates by themselves as a management tool, you will use them in conjunction with other management tools, such as the Azure portal.

Bringing It All Together

All the Azure management options available to you can be confusing when you first start working with Azure. You can clear up the confusion by understanding some key concepts, which will also help you be successful in answering management tool questions in the AZ-900 exam. These concepts are as follows:

- Azure PowerShell and the Azure CLI both provide essentially the same capabilities and enable you to manage most aspects of Azure through scripting. The difference is in **syntax**. If you are already familiar with PowerShell, then Azure PowerShell is the best solution for you. If you are most comfortable in the Linux Bash shell, the Azure CLI is the best solution for you.
- Azure PowerShell and the Azure CLI can both be **run natively** when installed on a device. These include Windows, Linux, and Mac devices.
- Azure PowerShell and the Azure CLI can both be run in a web interface through the Azure Cloud Shell. So, think of the Cloud Shell as just a **web-based means** to run either PowerShell or the Azure CLI when running them natively isn't possible or feasible, such as from a device that doesn't support them natively.
- The Azure Mobile App gives you **quick access** to many Azure management functions on your Android or iOS mobile device but is generally not as useful for complex management tasks. If you are out to dinner at a restaurant and need to reset a VM, the Azure Mobile App is the right solution. If you need to script the deployment of several resources, then Azure PowerShell or the Azure CLI running either natively or through the Azure Cloud Shell is the appropriate solution.

Creating and Managing Resources

Now that you have some understanding of the tools you can use to create and manage Azure resources, let's put that knowledge to work and create some. As you have learned in other chapters, you cannot create Azure resources without an Azure subscription. So, if you do not already have one, the first step is to create a free subscription. The following section explains how to do so.



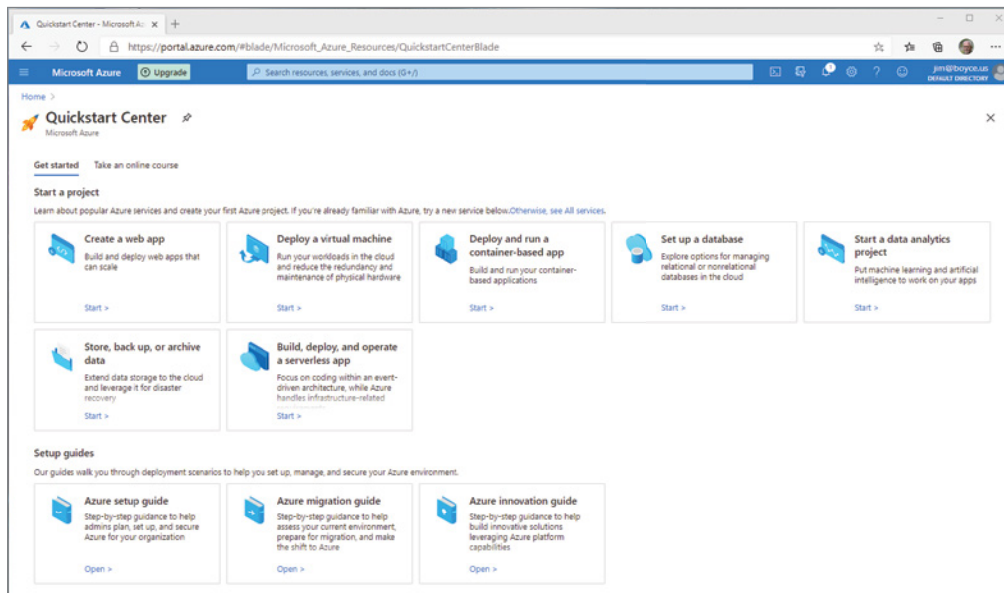
Although you could use Azure PowerShell or the Azure CLI to create resources, doing so requires some skill with those command environments. The following sections focus on using the Azure portal to create and manage resources.

Creating a Free Subscription

Creating a free Azure subscription is very easy. However, free subscriptions are available only to new users and **limited to one per person**, so you will need to sign in with an account that is not already associated to an Azure subscription. Start by navigating to <https://azure.microsoft.com/en-us/free> in a web browser. Then, click Start Free. You will need to log in with a Microsoft account. If you don't already have one, visit <https://account.live.com> to create one.

The Azure website will step you through the process of signing up for your free subscription. You will provide contact information and enter a credit card number. Microsoft won't charge anything on the card unless you eventually upgrade to a paid subscription later. You will see the Quickstart Center (see Figure 7.5) after you complete the sign-up process. You can get started with the options shown on the page or click the button in the upper-left corner of the browser (three horizontal lines, sometimes called the hamburger button) to view the Azure portal navigation options. The resulting menu is also referred to as the Azure portal menu.

FIGURE 7.5 The first thing you see after creating a free subscription is the Quickstart Center.



Creating Resource Groups

Before you create some Azure resources, you should create a resource group to contain them. Follow these steps to create a resource group:

1. Log into `portal.azure.com` with the account you used to create your free subscription.
2. In the Azure portal, click Resource Groups in the Navigate group to open the page shown in Figure 7.6.

FIGURE 7.6 Use the Resource Groups page to create and manage resource groups.

The screenshot shows the 'Create a resource group' page in the Azure portal. The page has a blue header with the Microsoft Azure logo and a search bar. Below the header, there's a breadcrumb trail: 'Home > Resource groups > Create a resource group'. The main content area is titled 'Create a resource group' and has a close button (X) in the top right corner. There are three tabs: 'Basics', 'Tags', and 'Review + create'. The 'Basics' tab is active. Below the tabs, there's a description of a resource group: 'Resource group - A container that holds related resources for an Azure solution. The resource group can include all the resources for the solution, or only those resources that you want to manage as a group. You decide how you want to allocate resources to resource groups based on what makes the most sense for your organization. [Learn more](#)'. Below this, there are two sections: 'Project details' and 'Resource details'. In 'Project details', there are two dropdown menus: 'Subscription' (set to 'Azure subscription 1') and 'Resource group' (set to 'myRG1'). In 'Resource details', there is a dropdown menu for 'Region' (set to 'US North Central US'). At the bottom of the page, there are three buttons: 'Review + create' (highlighted in blue), '< Previous', and 'Next: Tags >'. The browser's address bar shows the URL: 'https://portal.azure.com/#create/Microsoft.ResourceGroup'.

3. Click Create.
4. On the Create A Resource Group page, enter the name **myRG1** for the resource group.
5. Choose the region in which to create the resource.
6. Click Next.
7. On the Tags page, enter **RGType** as the name of the tag and **TestRG** as the value.
8. Click Review And Create.
9. If all is correct, you should see a message on the resulting page that validation passed. If not, navigate back in the browser to correct any issues.
10. Click Create to create the resource group.

Now when you navigate to the Resource Groups page you will see the group myRG1 that you just created. Next, let's add some resources to the resource group. The following section takes you through the process for creating a small number of different Azure resource types to familiarize you with the process and concepts.

Creating Azure Resources and Services

In this section you will create a handful of Azure resources of different types. Let's start with a virtual machine.

Creating a Storage Account

Creating a storage account is the first step in using Azure storage. One property of a storage account is its name. The storage account name **must be unique** across all Azure storage accounts, which means you might need to be creative to come up with a unique value. Use the following process to create a storage account in your free Azure subscription:

1. Log into the Azure portal and click Storage Accounts in the Azure Services group, or if you don't see Storage Accounts as an option, click in the search box and search for **storage account**.
2. On the resulting web page, click the Create Storage Account button.
3. Azure asks you to choose the subscription and resource group for the storage account (Figure 7.7). Click the Resource Group drop-down and choose the myRG1 resource you created previously.

FIGURE 7.7 Enter information to create your first storage account.

The screenshot displays the 'Create storage account' wizard in the Azure portal. The browser address bar shows the URL <https://portal.azure.com/#create/Microsoft.StorageAccount>. The page title is 'Create storage account'. Below the title, there are tabs for 'Basics', 'Networking', 'Data protection', 'Advanced', 'Tags', and 'Review + create'. The 'Basics' tab is active. A brief description of Azure Storage is provided. The 'Project details' section requires selecting a subscription and resource group. 'Subscription' is set to 'Azure subscription 1' and 'Resource group' is set to 'myRG1'. The 'Instance details' section contains several configuration options: 'Storage account name' is an empty text input; 'Location' is set to '(US) North Central US'; 'Performance' has 'Standard' selected; 'Account kind' is set to 'StorageV2 (general purpose v2)'; and 'Replication' is set to 'Read-access geo-redundant storage (RA-GRS)'. At the bottom, there are three buttons: 'Review + create', '< Previous', and 'Next: Networking >'.

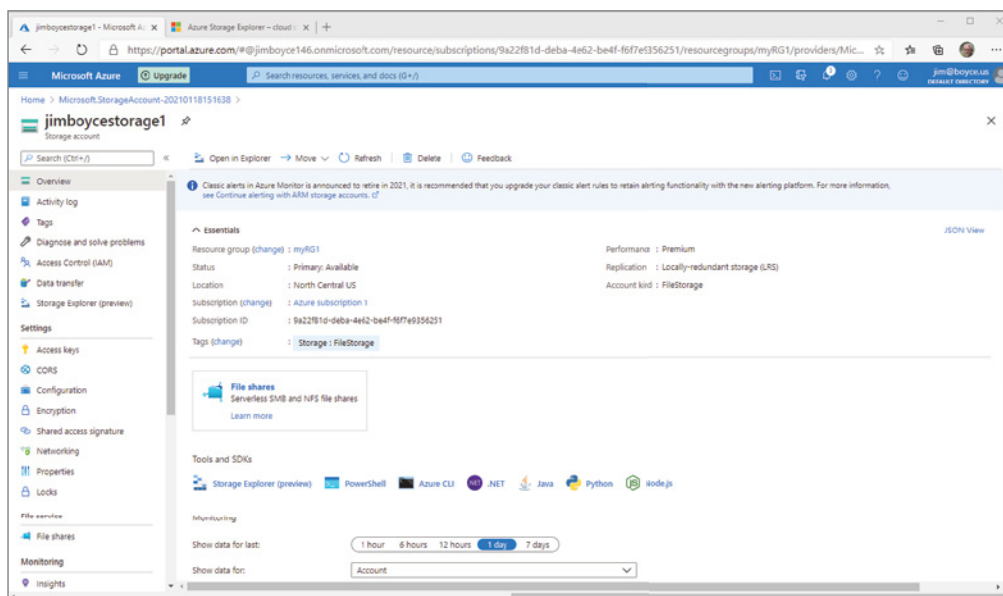
4. Enter a unique storage account name. You can use **only lowercase letters and numbers**.
5. For this example, choose Premium.
6. From the Account Kind list, select FileStorage.
7. From the Replication list, choose Locally-Redundant Storage (LRS).
8. Click Next.

At this point in the process, you must decide on the networking options for your storage account. Because this is a sample account you will use for testing, let's assume you won't put any information in Azure storage that poses any security or data privacy concerns. We will use the default options:

1. Keep the default settings of Public Endpoint and Microsoft Network Routing, then click Next.
2. Review the options for information but leave all of them unselected and click Next.
3. Again, review the options for informational purposes but leave them as is and click Next to add a tag to your storage account.
4. Enter **Storage** as the tag Name and **FileStorage** as the Value, and then click Review + Create.
5. If validation doesn't pass, correct any issues. Then click Create to create the storage account.

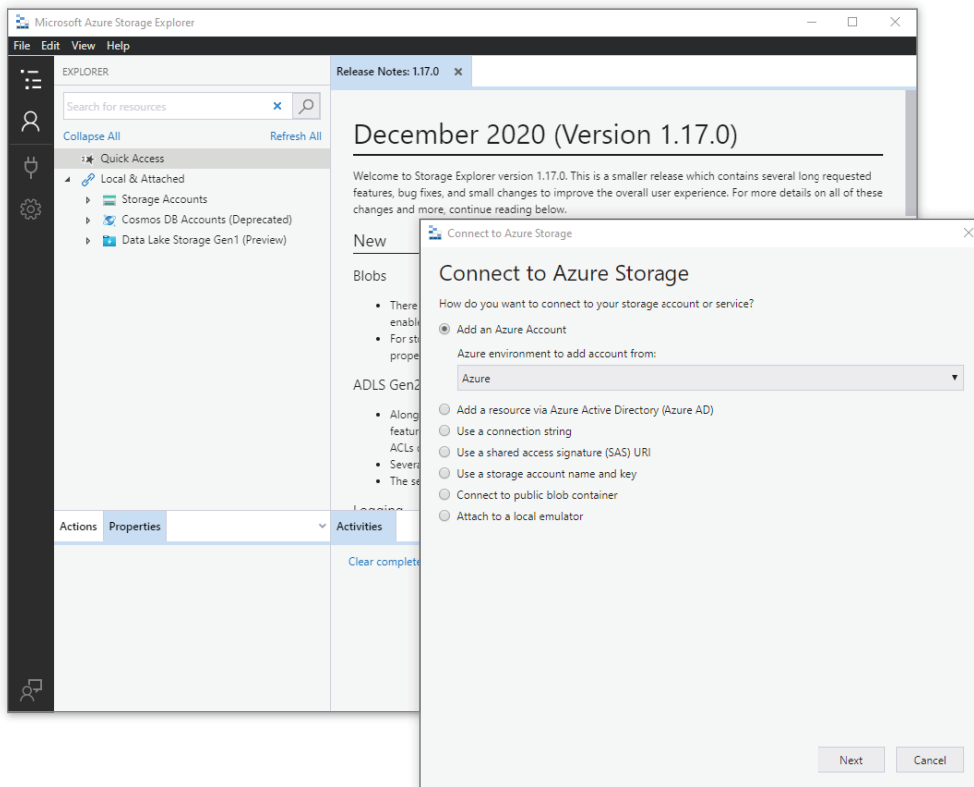
The Azure portal displays some status information as it creates the storage account. When the portal indicates that the deployment is complete, click Go To Resource. You should see something similar to Figure 7.8.

FIGURE 7.8 A storage account created in Azure called jimboycstorage1



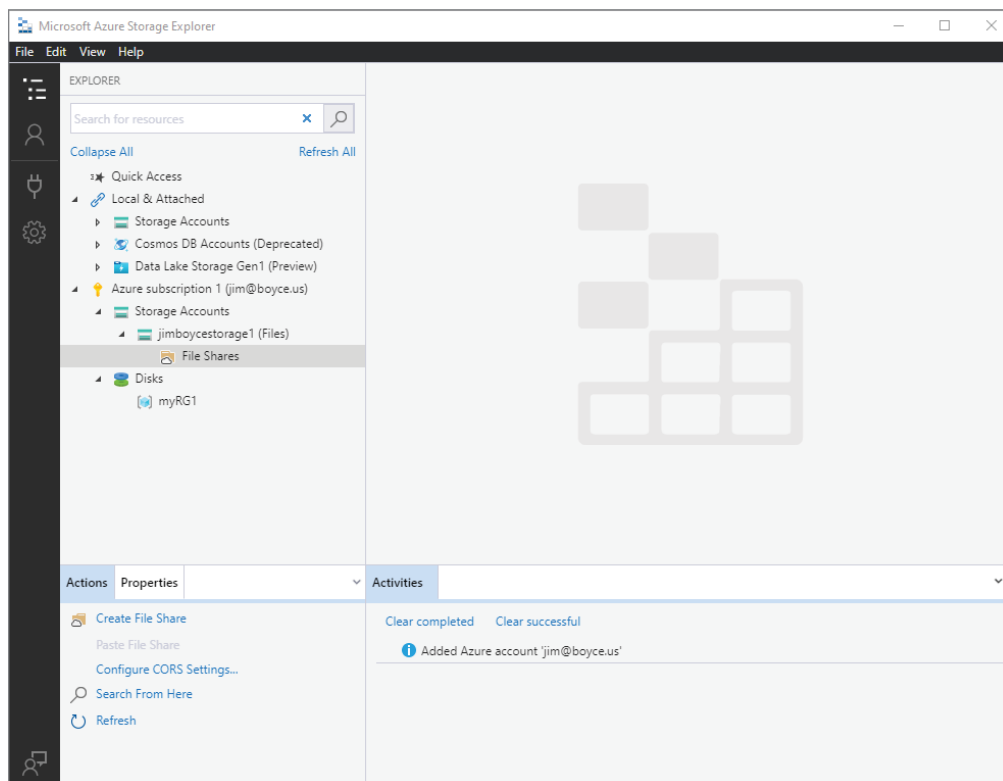
In this example, you can work with the storage as if it were local storage. First, however, you must download and install the **Azure Storage Explorer**. Click the download link on the page to download and install the version appropriate to your operating system. The first time you run the Azure Storage Explorer it prompts you to add storage (Figure 7.9). In this case we will add an Azure account. With **Add An Azure Account** selected, choose the environment where your Azure account resides (likely, Azure). Then click **Next**. When prompted, log in with the appropriate account. The Azure Storage Explorer adds the account.

FIGURE 7.9 Use the Azure Storage Explorer to view storage in Azure.



With your Azure account added, let's add a file share:

1. In the Azure Storage Explorer, select **File Shares**, as shown in Figure 7.10, and then click **Actions, Create File Share**.
2. Enter **test1** for the file share name and press **Enter**.
3. Click **New Folder** and create a folder called **Documents**.
4. Click **New Folder** and create another folder named **PowerShell**. Figure 7.11 shows the results.

FIGURE 7.10 An Azure account added to the Azure Storage Explorer

Although we won't walk through the process now, you can now upload files to the folders that you create. Just click Upload to select a file and upload it to Azure.

Creating a VM

One of the resource types you will probably use quite a bit in Azure is virtual machines. As with many Azure resource types, it's relatively easy to create a VM. Follow these steps to create a VM:

1. In the Azure portal, choose Virtual Machines from the navigation menu.
2. On the resulting Virtual Machines page, click Add. At this point you can either create a **customized VM** or start from a **preset configuration**. For simplicity at this point, choose Start With A Preset Configuration.
3. The Azure portal then offers several options, as shown in Figure 7.12. You can choose between Dev/Test and Production (the default option). Leave Production selected and choose General Purpose (D-Series).

FIGURE 7.11 Two folders created in a file share called test1

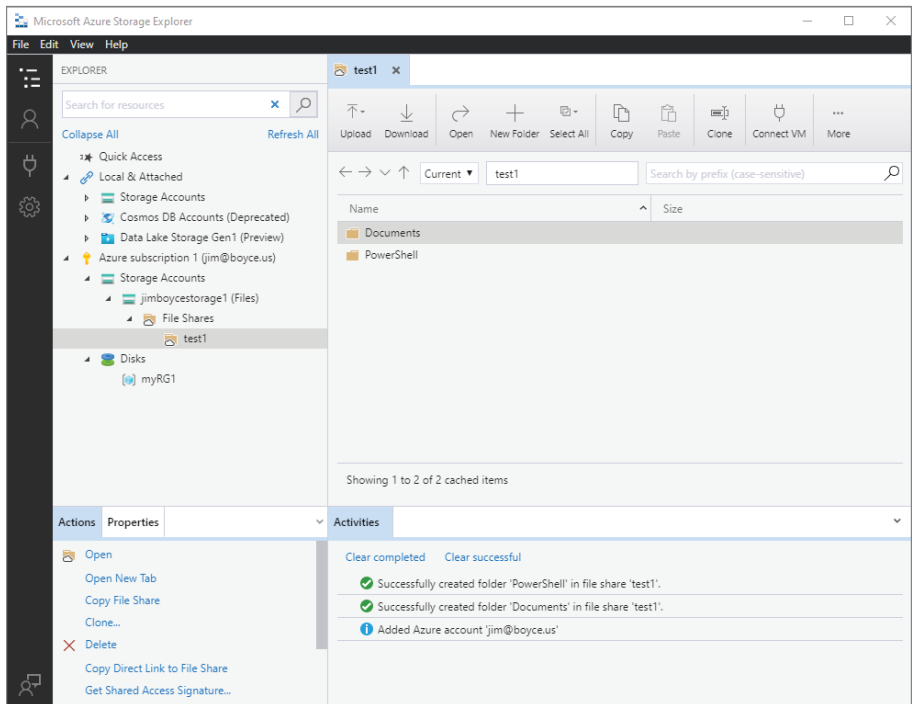
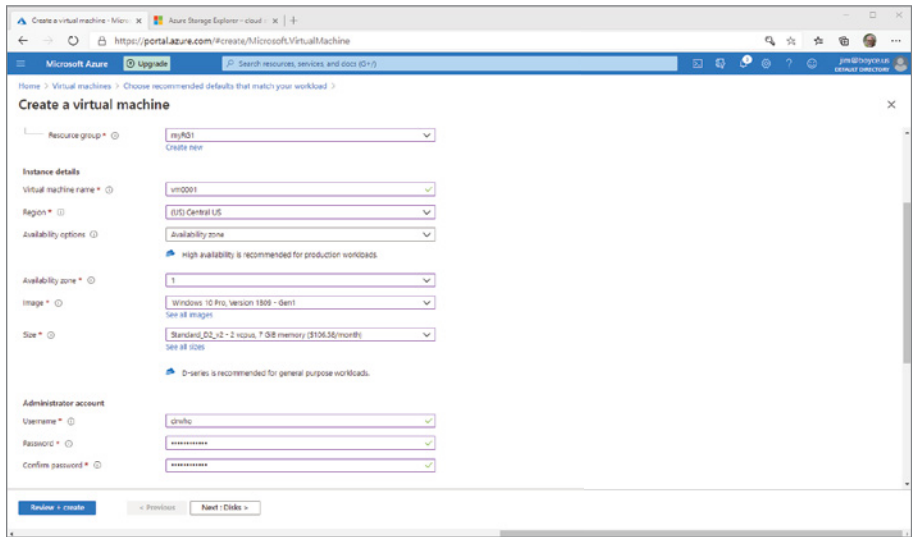


FIGURE 7.12 You can choose among several options when creating a preconfigured VM.



4. Click the button labeled Continue To Create A VM.
5. You could create a separate resource group for the VM, but in this case, select your previously created myRG1 group.
6. Enter **vm0001** as the VM name.
7. Choose an appropriate region (some regions do not provide the option to specify an availability zone).
8. If desired, choose an availability zone.
9. Choose the desired OS from the Image list.
10. Choose the VM size from the Size list, noting the monthly cost associated with the selection.

Although this is a free subscription, resources count against the subscription limits. For example, a VM might have a monthly cost of \$160 or so, which if it were running constantly would generate that amount in consumption cost. In this situation, however, you will create the VM, use it for a little while, and then delete it, which will stop any consumption costs for the VM.

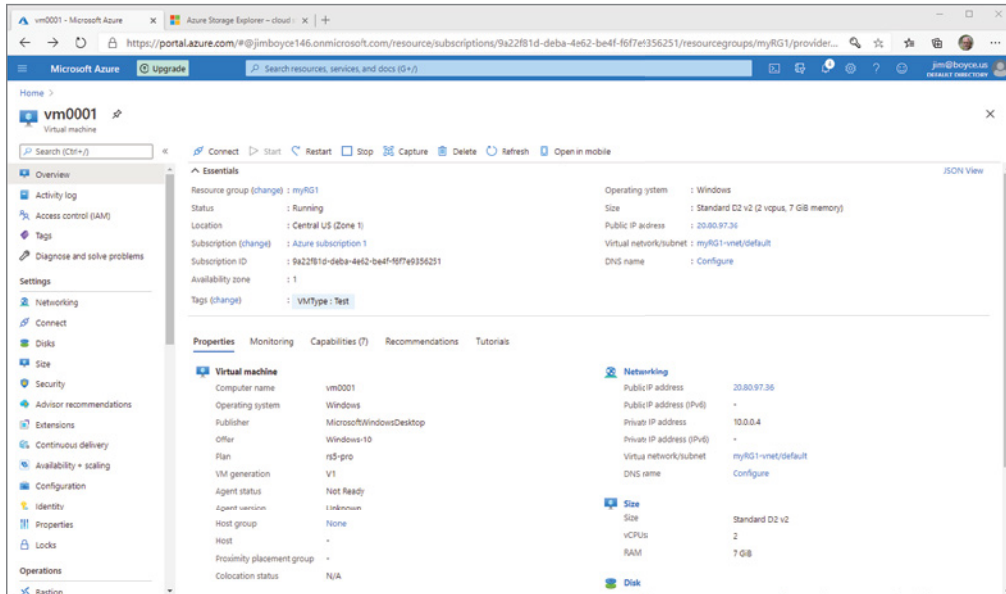
1. Enter an admin account name and password and click Next.
2. Review the default options on the Disks page and then click Next to use the defaults.
3. Review the default options on the Networking page and click Next to use the defaults.
4. Click through the Management and Advanced pages, accepting the defaults.
5. On the Tags page, add the name tag **VMType** with a value of **Test**. Note that the tag will be applied to 12 resource types used by the VM, and then click Review + Create.

At this point you might receive a validation error. For example, if you selected Windows 10 Pro as the OS image, there is a check box on the Basics page that you must select to confirm that you have an appropriate license for Windows 10 that you can use in Azure. Clear up any validation issues and then click Create to create the VM.

After the VM is created, you can click Go To Resource in the Azure portal to view its status. Figure 7.13 shows an example. Creating the VM is really just the first step in most cases. As Figure 7.13 indicates, for example, you can **configure a DNS name** for the VM and set other properties. You can also connect to the VM, restart it, stop it, and so on. If you would like to experiment with the VM, click Connect and follow the prompts to connect to and log into the VM. When you're finished experimenting, come back to the Azure portal and stop the VM by clicking Stop in the menu at the top of the page.



Why stop the VM? If you aren't using it, stopping the VM will pause most consumption charges associated with the VM. Even though this is a free Azure subscription, you don't want to use any more of your monthly credit than necessary.

FIGURE 7.13 Viewing status of a VM in the Azure portal

Creating a SQL Database

Next, create a SQL Database instance to see how easy it is to create one. You do not need to be a SQL expert here. To start the process, open the Azure portal and then click SQL Database in the navigation pane. On the resulting SQL Databases page, click Add to view the Create SQL Database page.

Follow these steps to complete the process of setting up a SQL database instance:

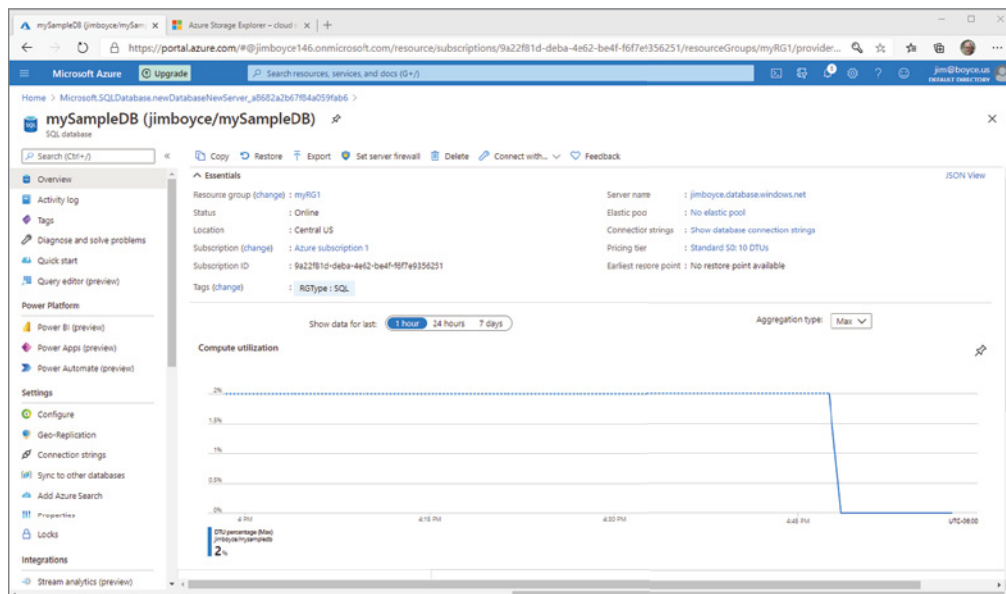
1. Choose the resource group you created earlier.
2. Enter **mySampleDB** in the Database Name field.
3. From the Select A Server drop-down, select Create New.
4. Enter a unique server name, admin credentials, and region, as shown in Figure 7.14, and then click OK.
5. Click Next to view the Networking page.
6. Since you won't be putting any data in the database for this example, choose No Access and click Next.
7. Review the default options for information and then click Next to add tags.
8. Choose **RGType** as the tag name and enter **SQL** as the value, and then click Review + Create.
9. Note the estimated operation cost for the resource and then click Create.

After Azure deploys the database, you can click Go To Resource to view the status. Figure 7.15 shows an example.

FIGURE 7.14 Use the Create SQL Database page to create an instance of a SQL database.

The screenshot shows the 'Create SQL Database' page in the Azure portal. The page is divided into several sections: 'Project details', 'Database details', and a 'New server' sidebar. In the 'Project details' section, the 'Subscription' is set to 'Azure subscription 1' and the 'Resource group' is 'myRG1'. In the 'Database details' section, the 'Database name' is 'mySampleDB' and the 'Server' is 'Select a server'. A red error message states 'The value must not be empty.' and a prompt says 'Please select a server first.' The 'New server' sidebar contains fields for 'Server name' (jimboyce), 'Server admin login' (druho), 'Password' (masked), 'Confirm password' (masked), and 'Location' (US Central US). At the bottom, there are buttons for 'Review > create' and 'Next: Networking >'. The 'Compute + storage' section has a link to 'Configure database'.

FIGURE 7.15 The Azure portal shows the status of a newly created SQL Database instance.




Obviously, additional considerations are involved in creating a SQL Database instance. For example, you will likely need access to it for management purposes and to connect it to other services. With a focus specifically on how to create the resource, we won't cover those topics here.

Deleting Resources and Services

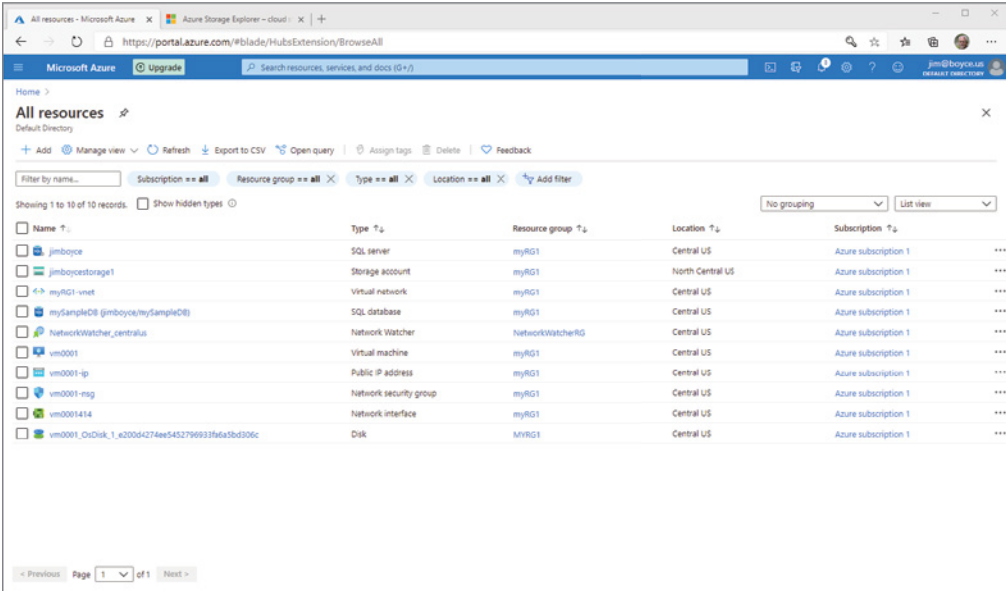
Knowing how to delete an Azure resource is as important as knowing how to create them. Although you are probably working here from a free subscription, you want to **avoid reducing your monthly credit** so that you have credit available to explore other features. As with the previous section, the focus in this section is on using the Azure portal to delete resources. Let's start by deleting the SQL Database instance you created in the previous section.

1. Open the Azure portal and navigate to the resource you want to delete.



The Recent Sources group in the Azure portal shows the resources you've used recently. You can also click All Resources in the navigation pane to view your resources, as shown in Figure 7.16.

FIGURE 7.16 Use All Resources to view and manage your Azure resources.



2. In All Resources, click the check box beside the mySampleDB that you created earlier and then click Delete in the top menu.

3. The Azure portal displays a warning message indicating that deleting the resource is irreversible. Click in the Confirm Delete box and type yes, then click Delete. Azure then deletes the database.



If you have problems finding a specific resource in All Resources, try sorting the view by Type.

Next, let's delete the VM that you created earlier:

1. In the Azure portal, click All Resources in the navigation pane and click vm0001. The Azure portal displays a page showing the VM's properties and status.
2. Click Delete. The Azure portal displays, indicating that the VM will be deleted but associated resources will not be.
3. Click Yes to delete the VM.
4. Delete the other resources with names starting with vm0001 to remove them as well, starting with the virtual network interface. You will then be able to delete the NSG and IP address associated with the VM.
5. Delete the virtual disk associated with the VM.

Summary

This chapter explored the tools available for managing Azure resources. You learned about Azure PowerShell and the Azure CLI, and how they can be installed natively or used within the Azure Cloud Shell. The chapter also explored the Azure Mobile App and how ARM templates fit into the bigger Azure management picture.

This chapter also introduced you to creating and deleting Azure resources so that you can begin experimenting more deeply with Azure. As you add resources, just keep in mind that even if you are using a free subscription, those resources will reduce your monthly credit. As explained throughout this book—particularly in Chapter 6, “Azure Pricing, Service Levels, and Lifecycle”—use best practices to turn off Azure resources you aren't using. This is particularly true for VMs that will incur consumption costs as long as they are running, even if you aren't using them.

Exam Essentials

Describe the functionality and use of the Azure portal, Azure PowerShell, the Azure CLI, the Cloud Shell, and the Azure Mobile App. Although the Azure management tools can seem confusing at first glance, once you understand some basic concepts the relationship

between the tools makes sense. Azure PowerShell and the Azure CLI provide essentially the same management capabilities, with both providing command-line, scripted management of all Azure resources and functions. The main difference between the two is in syntax. Both can run natively after you install them on a device, or you can run either one in the Azure Cloud Shell.

The Azure Cloud Shell is a web-based environment specifically for running either Azure PowerShell or Azure CLI command sessions.

The Azure Mobile App gives you some management capabilities on Android and iOS mobile devices but naturally doesn't provide the same management capabilities as the Azure portal, Azure PowerShell, or the Azure CLI.

If you need to perform relatively simple, one-off management tasks, the Azure portal is often the simplest choice. If you need that capability from a mobile device, then the Azure Mobile App is a good solution. For anything requiring complex actions, Azure PowerShell or the Azure CLI are most appropriate. When you need to build out repeatable processes for deploying and managing resources with related resources, ARM templates are the appropriate choice.

Review Questions

1. You are a developer and need to deploy a single VM to test some code overnight. Which of the following management options is the best choice?
 - A. Azure PowerShell
 - B. Azure CLI
 - C. Azure portal
 - D. ARM templates
2. You are an infrastructure management engineer for your organization. You work most with Linux servers and are therefore very familiar with Bash and know what command you would use to retrieve specific properties of a VM. Which of the following management solutions should you use to retrieve a property of one of the VMs that you manage?
 - A. Azure PowerShell
 - B. Azure CLI
 - C. Azure portal
 - D. Azure Monitor
3. You are the Director of IT for your organization. Your CIO has asked you what methods or processes your organization should consider using to build out the capability to deploy large numbers of Azure resources with connected services and policies in a controlled way. Which of the following solutions should you recommend?
 - A. Azure PowerShell
 - B. Azure CLI
 - C. Azure portal
 - D. ARM templates
4. You want to use the Azure CLI to perform management functions in Azure. Which of the following configurations does not suit this requirement?
 - A. You install the Azure CLI on a Windows 10 PC.
 - B. You install the Azure CLI on a Mac.
 - C. You open the Azure portal in a browser and then run the Azure CLI in the Azure Cloud Shell.
 - D. You run the Azure CLI from PowerShell on a Windows 10 device.
5. You are out of the office at an appointment and one of your team members sends you an urgent request to reset a web app because the only other people with the appropriate access are not available. All you have is your mobile device. Which two options can you use to reset the web app?
 - A. Azure portal
 - B. Azure CLI

- C.** Azure Cloud Shell
 - D.** Azure PowerShell
- 6.** Which blade in the Azure portal should you use to view security alerts and recommendations?
- A.** All Services
 - B.** App Security
 - C.** Security Center
 - D.** Monitor